

Entorno para el acceso remoto vía web a laboratorios multicluster

Adrián Pousa¹, Armando De Giusti², Marcelo Naiouf³
{apousa, degiusti, mnaiouf}@lidi.info.unlp.edu.ar

Instituto de Investigación en Informática (III-LIDI) – Facultad de Informática – UNLP

Abstract

This paper presents the research and development of an environment which allows remotely accessing multicluster Labs. The result is a management software layer that can be accessed through a Web application, allowing the remote operation of a multicluster architecture both to run programs over it and to manage it.

The Web application gives students, teachers and researches the possibility of working remotely, in a transparent fashion, assigning them a working space and allowing them to compile and run programs with different levels of interaction.

The developed software is analyzed so its use by students and researchers of concurrent, distributed and parallel processing in the UNLP.

Keywords: *Distributed systems. Clusters. Parallel Processing. Web Systems. Operating System middleware.*

Resumen

El trabajo presenta la investigación y desarrollo de un entorno que permite el acceso remoto a laboratorios multicluster. El resultado es una capa de software de administración que puede ser accedida mediante una aplicación Web, permitiendo la operación remota de una arquitectura multicluster tanto para ejecutar programas sobre ella como para administrarla.

La aplicación Web da la posibilidad a alumnos, docentes e investigadores de trabajar en forma remota sobre una arquitectura de este tipo de manera transparente, asignándoles un espacio de trabajo y permitiendo compilar y ejecutar programas con distintos niveles de interacción.

Se analiza el software desarrollado así como su utilización por estudiantes e investigadores de procesamiento concurrente, distribuido y paralelo en la UNLP.

Palabras Clave: *Sistemas Distribuidos. Clusters. Procesamiento Paralelo. Sistemas Web. Software de Sistema Operativo.*

WARSO. Workshop de Arquitectura, Redes y Sistemas Operativos

¹ Becario Telefónica e III-LIDI. Docente Auxiliar. Facultad de Informática UNLP.

² Investigador Principal CONICET. Profesor Titular D.E. Facultad de Informática UNLP.

³ Profesor Titular D.E. Facultad de Informática UNLP.

1 INTRODUCCION

La utilización de redes de computadoras como maquinas paralelas virtuales se ha generalizado y el desarrollo de software de administración de recursos físicos, comunicaciones y software disponible en clusters, multiclusters y grids es un tema actual de investigación y desarrollo.

Sintéticamente, un cluster es un tipo de sistema paralelo que consiste de un conjunto de computadoras independientes, conectadas por alguna forma de red, y que configuran un recurso de procesamiento integrado [1] [2] [3]. Actualmente los clusters constituyen una clase de procesadores de alto rendimiento muy utilizados, al punto que más del 60% de las computadoras que figuran en el ranking de las “top-500” son clusters[4]. Cuando se conectan dos o más clusters sobre una red tipo LAN o WAN, se tiene alguna forma de multicluster. Existen diferentes tipos de multicluster según las características de los procesadores, el sistema operativo que utiliza cada uno y la red que los conecta [16]. Básicamente cada uno de estos aspectos define un grado de heterogeneidad en la configuración global, que será muy importante al estudiar performance [5] [17].

Un Grid es un tipo de sistema distribuido que permite compartir e integrar recursos distribuidos geográficamente, incluyendo procesadores, instrumentos, bases de datos e incluso recursos humanos. Esta clase de arquitectura permite definir múltiples configuraciones colaborativas entre comunidades de usuarios, donde cada usuario “ve” una arquitectura única [6] [7] [8].

En la Facultad de Informática existen varios subsistemas en red que son utilizados por alumnos, docentes e investigadores como maquinas paralelas virtuales: Cluster de 16 maquinas (CL1), Cluster IBM de 20 máquinas (CL2) y Cluster S101 de 16 máquinas (CL3). Además se realizan investigaciones de procesamiento paralelo empleando en conjunto todas las máquinas de CL1, CL2 y CL3. Por otra parte (dentro de los proyectos de I/D del III-LIDI [9]) estos clusters pueden trabajar “en Grid” con redes de procesadores de otras Universidades.

Los clusters están formados por distintos equipos heterogéneos en cuanto a los recursos que disponen, como cantidad de memoria y capacidad de almacenamiento y procesamiento.

El acceso a la arquitectura multicluster disponible en la Facultad requería que los usuarios trabajaran en el lugar físico donde se encuentran los clusters o bien remotamente a través de protocolos que requieren habilitar reglas de firewalls, que no dan al usuario una vista transparente y que pueden tener problemas de seguridad que se pueden evitar.

Por estas razones se decidió analizar e implementar una capa de software que permita ser accedida vía Web, brindando mayor transparencia e interacción y dando a los usuarios un espacio de trabajo sobre una arquitectura multicluster de manera que puedan compilar, ejecutar programas y obtener resultados. Por otra parte, la aplicación planteada da la posibilidad de administración remota de la arquitectura multicluster.

La capa de software se desarrolló en JAVA y se ubica por encima de los entornos de ejecución sobre arquitecturas multicluster como es el caso de los entornos MPI que proveen librerías para programación distribuida.

El trabajo está organizado como sigue: la Sección 2 define el aporte del trabajo. La Sección 3 plantea el problema de la ejecución remota. La Sección 4 plantea el esquema de solución y el entorno desarrollado, mientras la Sección 5 analiza el caso particular del acceso a los laboratorios multicluster. Finalmente, las Secciones 6 y 7 muestran algunas conclusiones y líneas de trabajo futuras respectivamente.

2 APOORTE DEL TRABAJO

El aporte de este trabajo lo constituye el análisis y desarrollo de una capa de software para la administración y utilización de una arquitectura multiclustero; esta capa de software puede ser accedida vía Web y brinda a los usuarios una vista transparente de la arquitectura.

3 EJECUCION REMOTA DE PROGRAMAS

La Facultad de Informática cuenta con varios clusters que son utilizados por docentes, investigadores y alumnos. El acceso a dicho equipamiento se realizaba de dos maneras: local y remotamente mediante el protocolo SSH a manera de consola de texto.

El acceso local requiere que el usuario trabaje en el lugar físico donde se encuentra el cluster, lo que restringe la disponibilidad. El acceso remoto mediante SSH si bien permite utilizar el cluster toda vez que el acceso es permitido, tiene como inconvenientes que no es transparente, que se requieren controles a nivel de firewall y se pierde seguridad.

Interesa entonces un acceso remoto, con seguridad y transparencia para los usuarios desde un entorno interactivo amigable. Descartados SSH y Telnet por las razones mencionadas en el párrafo anterior el trabajo se enfocó en el desarrollo de una capa de software accesible vía Web, que brinde al usuario una vista transparente y un espacio de trabajo para que éste opere el sistema de acuerdo con los derechos que le han sido asignados (básicamente ejecutar programas sobre el cluster y transferir los resultados).

4 SOLUCION PROPUESTA Y ENTORNO DESARROLLADO

La aplicación Web desarrollada brinda la posibilidad a alumnos, docentes e investigadores de trabajar en forma remota sobre una arquitectura multiclustero de manera transparente, asignándoles un espacio de trabajo y permitiendo compilar y ejecutar programas con distintos niveles de interacción. En este trabajo se analizan las funcionalidades del software desarrollado para el acceso remoto a los Clusters de la Facultad de Informática de la UNLP, utilizados por varias cátedras de grado y postgrado, en particular las relacionadas con procesamiento concurrente, distribuido y paralelo.

El esquema de solución propuesto se muestra en la Figura 1. La capa de software (entorno) está implementada en JAVA y es accesible vía Web a través de un servidor Tomcat. Esta capa interactúa con los entornos de ejecución basados en MPI en dos de sus varias implementaciones (LAM y MPICH2).

De esta forma es posible administrar estos entornos a través de la aplicación Web, compilar programas en lenguaje C utilizando las librerías de programación distribuida provistas por los entornos MPI, además de ejecutar dichos programas y controlar las ejecuciones realizadas.

Se utiliza además un motor de base de datos MySQL para almacenar aspectos de configuración, los usuarios de la aplicación y para mantener registros sobre las ejecuciones hechas. Todas estas capas de software funcionan bajo un sistema operativo Fedora.

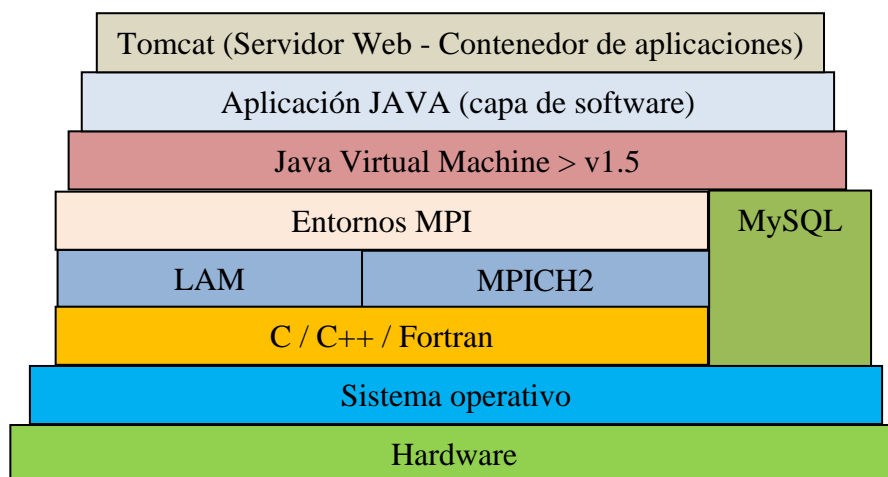


Figura 1. Esquema en capas

Es importante mencionar que todas las herramientas utilizadas tanto para el desarrollo de la capa de software como para la utilización de la arquitectura multicluster son de libre distribución y gratuitas.

En relación a la seguridad, en esta instancia se tomaron sólo medidas mínimas y es tema de trabajo futuro el caso de la seguridad en este tipo de arquitecturas.

El sistema permite ejecutar remotamente una aplicación que no se sabe que tipo de código pueda contener (esto incluye la posibilidad que exista código malicioso que pueda dañar todo el sistema).

Por lo tanto, toda la aplicación Web, incluyendo el servidor y la estructura de archivos para cada usuario, trabajan bajo un único usuario de sistema sin privilegios de administrador. Esto permite que no se puedan modificar estructuras de sistema sin permisos (pero no evita la lectura de información o el borrado de archivos sobre los que el usuario tenga permisos). En particular, por las características del sistema, cada usuario de aplicación tiene el mismo usuario de sistema y por lo tanto tiene permisos para modificar la estructura de archivos de otros usuarios de aplicación. Igualmente constituye una medida mínima para evitar daños mayores.

Toda ejecución de usuario está registrada y la aplicación Web provee un mecanismo transparente sobre la estructura de archivos que le impide obtener información extra a la que ya tiene en su espacio de trabajo, y además cada usuario está identificado lo que permite registrar cualquier operación incorrecta.

5 UTILIZACION PARA EL ACCESO A LOS LABORATORIOS MULTICLUSTER

5.1 Arquitectura

En la Facultad de Informática existen varios subsistemas de procesamiento en red que son utilizados por alumnos, docentes e investigadores como maquinas paralelas virtuales, tal como se describió en la Sección 1. Estos clusters están formados por distintos equipos heterogéneos en cuanto a los recursos que disponen como ser: cantidad de memoria, capacidad de almacenamiento y procesamiento.

Un usuario que quiera ejecutar un programa escrito en C usando la librería MPI deberá seguir los siguientes pasos:

- Conectarse remotamente vía Web a la aplicación Web escrita en JAVA
- Enviar su programa
- Compilar y pedir la ejecución del programa a la aplicación Web

Gracias a la capacidad de JAVA de ejecutar desde una aplicación escrita en este lenguaje otras aplicaciones o comandos, el usuario recibirá la salida de la ejecución de su programa C/MPI.

Desde el punto de vista académico, la solución reduce el tiempo de configuración del cluster por el alumno/usuario: simplemente se conecta a la aplicación Web y se concentra en la ejecución y evaluación de su aplicación.

5.2 Características de la aplicación Web

Cada usuario dentro de la aplicación Web tiene un espacio de trabajo semejante a un sistema de archivos de cualquier sistema operativo.

Este espacio de trabajo permite a los usuarios (con un buen nivel de interacción gracias al uso de AJAX) crear, modificar y eliminar archivos o directorios, subir archivos, comprimirlos o descomprimirlos utilizando zip, editar archivos, visualizarlos, compilar código fuente y ejecutar el código compilado en los tres modos de ejecución disponibles.

El espacio de trabajo es independiente del sistema de archivos del sistema operativo, lo que brinda un aceptable nivel de transparencia. Incluye además las tareas de administración para los usuarios administradores.

5.2.1 Perfiles de usuario

La aplicación Web permite dos perfiles de usuarios:

- Usuarios Standard (fundamentalmente alumnos): tienen un espacio de trabajo que es equivalente a un sistema de archivos; en él pueden editar, compilar y ejecutar programas en los tres modos de ejecución disponibles además de detener estas ejecuciones.
- Usuarios Administradores (docentes): agregan a las posibilidades de los usuarios comunes las tareas de administración tales como realizar altas, bajas y modificaciones de usuarios, de equipos y clusters, cambiar modos de acceso al cluster, correr o detener ambientes de ejecución MPI y manejar registros de las ejecuciones hechas.

5.2.2 Compilación y modos de ejecución

La aplicación Web permite el uso de las librerías para programación distribuida MPI utilizando el lenguaje de programación C. Es posible compilar y ejecutar programas utilizando estas librerías en dos distribuciones LAM y MPICH2.[10] [11]

Una vez compilado un programa puede ser ejecutado utilizando tres modos de ejecución:

- Sincrónico: Se envía el pedido de ejecución al servidor y se espera una respuesta en lo inmediato, por lo tanto la salida de la ejecución es retornada en la misma página WEB.
- Asíncrono: Se envía el pedido de ejecución al servidor y no se espera una respuesta en lo inmediato. Es utilizado para aplicaciones que tienen una demora importante, por lo tanto la salida de la ejecución es retornada en un archivo dentro del espacio de trabajo del usuario, y el contenido de este archivo puede visualizarse posteriormente.
- Interactivo: Da al usuario la misma sensación que tiene cuando ejecuta una aplicación de este tipo en una consola, donde la salida de la ejecución se va mostrando línea a línea. El usuario envía a ejecutar el programa y el servidor va entregando de a línea o conjunto de líneas a medida que se desarrolla la ejecución, las cuales va mostrando en la página Web.

Los modos sincrónico e interactivo son similares; la diferencia es que el modo sincrónico envía el pedido de ejecución y espera toda la salida de programa completa, y luego esta salida es mostrada en la página Web. Si el programa fallara por algún motivo el usuario no sabría la causa de la falla o el momento en que el programa se detuvo, recibiendo sólo una pantalla de salida en blanco o un mensaje genérico de error.

En el caso del modo interactivo, por cada línea o conjunto de líneas que retorne el programa que se este ejecutando en el cluster, se irán mostrando una por una en la pagina Web, de esta forma si el programa se detuviera o fallara inesperadamente le dará una idea al usuario de dónde puede estar el error.

Las ejecuciones sobre una arquitectura de cluster, generalmente, suelen parecerse más al modo de ejecución asíncrono donde se envía a un programa ejecutar al cluster y se espera una salida en un archivo con una demora importante de tiempo (minutos, horas, unos pocos días).

Notar que las aplicaciones de tratamiento masivo de datos, que habitualmente se ejecutan en clusters suelen tener estos requerimientos. Se supone que el programa fue anteriormente probado, asegurando la ausencia de errores, y luego ejecutado en el cluster, por lo tanto “no debería fallar”.

Los modos sincrónico e interactivo de la aplicación surgen por el uso práctico que se le da en algunas cátedras de la carrera, donde los programas no suelen tener demoras importantes, son más bien simples y el uso del cluster es con fines educativos.

5.2.3 Modos de acceso

Una arquitectura de cluster es un recurso en el que dos usuarios no deberían estar ejecutando al mismo tiempo para que puedan aprovecharse al máximo todos los recursos y tengan sentido las mediciones de tiempo de ejecución que caracterizan la performance en los sistemas paralelos.

La aplicación Web permite establecer dos modos de acceso:

- **Exclusivo:** Se asigna a un usuario una franja horaria para que pueda trabajar impidiendo que otros usuarios puedan ingresar al sistema mientras se este haciendo uso de la arquitectura. No se les permite ningún tipo de acceso a otros usuarios (ni siquiera cuando el cluster esté ocioso), y no se les permite autenticación, salvo que sean administradores y que se deba solucionar algún problema para que el usuario que tiene asignado el uso exclusivo del cluster pueda usarlo sin inconvenientes.
- **Compartido:** A pesar del nombre, el modo compartido, no significa que se permita el uso del cluster a dos procesos de distintos usuarios al mismo tiempo, sino que se sigue manteniendo la exclusividad en la ejecución sobre la arquitectura de cluster.
Permite la autenticación de usuarios dejándolos operar sobre su área de trabajo, pero sólo les deja ejecutar programas si el cluster no esta ejecutando programas de otro usuario; de esta forma se aprovechan los tiempos ociosos de la arquitectura.
En caso de que el usuario desee ejecutar un programa sobre el cluster y el mismo esté ocupado, la aplicación reintentará la ejecución hasta que logre el acceso para ejecutar o bien hasta que el usuario cancele el pedido de ejecución.

Estos dos modos surgen de las necesidades de los usuarios, los alumnos suelen correr programas de muy corta duración (algunos minutos) que no requieren un tiempo exclusivo y que bien puede compartirse con otros alumnos; en cambio, los investigadores, suelen ejecutar programas de mayor duración (horas) y requieren exclusividad de la arquitectura por un determinado periodo de tiempo.

6 CONCLUSIONES

Se ha presentado un desarrollo de entorno y soporte de administración de recursos WEB, para la utilización de arquitecturas multiprocesador tipo cluster y multicluster en forma remota, brindando una interfaz transparente.

Este tipo de solución, que integra herramientas de uso libre, es particularmente importante en las carreras de Informática, donde un número importante de alumnos de grado y postgrado requieren trabajo experimental sobre arquitecturas multiprocesador disponibles en la Facultad.

7 LINEAS DE TRABAJO FUTURAS

Existen algunos aspectos que se están estudiando para agregar mayor funcionalidad a la aplicación Web. Uno de ellos es el de ejecutar programas utilizando un manejador de recursos y un scheduler como pueden ser Torque y Maui respectivamente [12] [13]. Esta combinación permite que los programas se encolen y sean ejecutados de acuerdo a alguna política de scheduler, además de una mejor utilización de los recursos.

Otro aspecto es el de fragmentar la arquitectura multicluster de manera de tener varios clusters para distintos propósitos y con diferentes políticas. Por ejemplo, varios alumnos pueden estar utilizando parte de un cluster de forma compartida mientras que al mismo momento un investigador puede tener parte del cluster de forma exclusiva.

Así como la arquitectura puede fragmentarse puede también expandirse incorporando clusters que tal vez no estén físicamente en el mismo lugar. Actualmente, se está trabajando en la conexión de clusters de otras universidades.

Un último aspecto es el de adaptar la capa de software para que funcione en una infraestructura de grid, donde se consideran equipos como nodos de grid distribuidos ya no en una LAN (como pueden considerarse los clusters) sino en una WAN, y cada nodo puede tener a su vez clusters conectados. Esta infraestructura proporciona mayor disponibilidad de recursos y puede suponer una mayor potencia de cálculo. Un objetivo es hacer funcionar la capa de software sobre middlewares de grid como son Globus Toolkit [14] o GLite[15].

REFERENCIAS

- [1] Grama A., Gupta A., Karypis G., Kumar V. "Introduction to Parallel Computing". Second Edition. Addison Wesley, 2003.
- [2] Jordan H, Alagband G. "Fundamentals of parallel computing". Prentice Hall, 2002.
- [3] Dongarra J, Foster I, Fox G, Gropp W, Kennedy K, Torczon L, White A. "The Sourcebook of Parallel Computing". Morgan Kauffman Publishers. Elsevier Science, 2003.
- [4] The TOP500 project <http://www.top500.org/>
- [5] Jiang, Yeung. "Scalable Inter-Cluster Communication System for Clustered Multiprocessors". 1997.
- [6] Foster I., Kesselman C., Kaufmann M. "The Grid 2: Blueprint for a New Computing Infrastructure". The Morgan Kaufmann Series in Computer Architecture and Design. 2 edition (November 18, 2003).
- [7] Juhasz Z. (Editor), Kacsuk P. (Editor), Kranzlmuller D. (Editor). "Distributed and Parallel Systems: Cluster and Grid Computing". The International Series in Engineering and Computer Science. Springer; 1 edition (September 21, 2004)
- [8] Berman F.(Editor), Fox G.(Editor), Hey A.(Editor). "Grid Computing: Making The Global Infrastructure a Reality". John Wiley & Sons (April 8, 2003).
- [9] III-LIDI <http://lidi.info.unlp.edu.ar>
- [10] LAM/MPI Parallel Computing <http://www.lam-mpi.org>
- [11] MPICH2/MPI Parallel Computing <http://www-unix.mcs.anl.gov/mpi/mpich2/>
- [12] Torque Resource Manager <http://www.clusterresources.com/pages/products/torque-resource-manager.php>
- [13] Maui Cluster Scheduler <http://www.clusterresources.com/pages/products/maui-cluster-scheduler.php>

[14] Globus Toolkit <http://www.globus.org>

[15] GLite <http://glite.web.cern.ch/glite>

[16] De Giusti A. et al “ Parallel algorithms on Multi-Cluster Architectures using GRID Middleware. Experiences in Argentine Universities” Proceedings of the I Iberian Grid Infrastructure Conference. Spain. Mayo 2007. Pag. 322-332

[17] Naiouf M. R., De Giusti L. C., Chichizola F., De Giusti A. E. “Dynamic Load Balancing on Non-homogeneous Clusters”. G.Min et al. (Eds.): ISPA 2006 Ws, LNCS 4331, pags. 65-73, 2006. Springer – Verlag. Berlin Heidelber 2006.